UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/692,436	10/22/2003	Avinash Sodani	42P17406	8405

8791 7590 11/13/2006

BLAKELY SOKOLOFF TAYLOR & ZAFMAN
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1030

EXAMINER	
HUISMAN, DAVID J	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 11/13/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/692,436	SODANI, AVINASH	
	Examiner	Art Unit	
	David J. Huisman	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *20 September 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-24* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,2,4-13,15-18,20,21,23 and 24* is/are rejected.

7)☒ Claim(s) *3,14,19 and 22* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *20 September 2006* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-24 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Amendment as received on 9/20/2006.

Drawings

3. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they include the following reference character(s) not mentioned in the description:

- In Fig.4, reference numbers 432, 434, and 440 do not appear in the specification.

Corrected drawing sheets in compliance with 37 CFR 1.121(d), or amendment to the specification to add the reference character(s) in the description in compliance with 37 CFR 1.121(b) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

5. Claims 1-2 are rejected under 35 U.S.C. 102(b) as being anticipated by Yeager et al., U.S. Patent No. 5,758,112 (herein referred to as Yeager).

6. Referring to claim 1, Yeager has taught a processor comprising:

a) a first register alias table including a first number of read ports to translate a first set of logical register addresses to physical register addresses. See Fig.3, components 204 and 352 (floating-point rename table with 16 read ports).

b) a second register alias table including a second number of read ports to translate a second set of logical register addresses to physical register addresses, wherein said first number that includes at least one read port for each source operand for an instruction is greater than said second number. See Fig.3, components 206 and 354 (integer rename table with 12 read ports). Also, see column 8, lines 55-59, and note that there is a read port for each source operand for an instruction. That is, among the 16 read ports in the floating-point rename table, four of those ports are assigned to an instruction, and among those four ports, each port is assigned to an operand (source or destination).

7. Referring to claim 2, Yeager has taught a processor as described in claim 1. Yeager has further taught that said first number is proportional to a third number of logical register addresses in said first set. See Fig.3 and column 8, lines 55-67, and note that a third number of addresses is

16 addresses (4 addresses (registers 358, 360, 362, and 364) in each of 4 parallel instructions). This is the reason 16 read ports are required (so that 4 instructions, each having 4 registers, may be renamed in parallel). Clearly, 1 set of 4 registers 4 read ports. So, it can be seen that 4 addresses is to 4 read ports, 8 addresses is to 8 read ports, 12 addresses is to 12 read ports, and 16 addresses is to 16 read ports. Consequently, the first number is proportional to a third number.

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9. Claims 4-13, 15-18, 20-21, and 23-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Yeager.

10. Referring to claim 4, Yeager has taught a processor as described in claim 1. Yeager has not taught a trace cache to supply a trace of micro-operations to said first register alias table and said second register alias table. However, Official Notice is taken that a trace cache and its advantages are well known and accepted in the art. Trace caches speed up the fetching process by caching instruction traces, which are sequences of decoded instructions. For example, a taken branch penalty is eliminated, since separate basic blocks appear contiguous in a trace. Also, since decoded instructions are stored in the trace cache, there is no need to decode them again next time they are fetched, which would save time. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Yeager to include a trace

cache that provides instructions to the first and second RATs. And, one would be motivated to make such a combination to achieve time savings in the fetching/decoding process.

11. Referring to claim 5, Yeager has taught a processor as described in claim 4. The modified Yeager has further taught that said trace cache includes trace cache logic to build said trace limiting a third number of live-in and live-out logical registers to said second number. Each group of 4 instructions executed in parallel from a given trace can only have up to 4 live-out registers (4 read ports) and 8 live in-registers (8 read ports).

12. Referring to claim 6, Yeager has taught a method comprising:

a) storing translations from logical register addresses to physical register addresses in a first register alias table. See Fig.3, components 204 and 352 (floating-point rename table with 16 read ports).

b) storing translations from logical register addresses to physical register addresses in a second register alias table, where said second register alias table has fewer read ports than said first register alias table that includes at least one read port for each source operand for an instruction. See Fig.3, components 206 and 354 (integer rename table with 12 read ports). Also, see column 8, lines 55-59, and note that there is a read port for each source operand for an instruction. That is, among the 16 read ports in the floating-point rename table, four of those ports are assigned to an instruction, and among those four ports, each port is assigned to an operand (source or destination).

c) Yeager has not explicitly taught that the translations stored in the first register alias table are frequently used translations while the translations stored in the second register alias table are less-frequently used translations. However, recall that Yeager's first table stores translations

corresponding to floating-point operations while Yeager's second table stores translations corresponding to integer operations. Consequently, if there are more floating-point instructions in a given program than integer instructions, then the floating-point translations would be considered to be frequently used translations with respect to integer translations, which would be less-frequently used. Clearly, any given program could include more floating-point instructions than integer instructions. For instance, in signal processing (DSP) applications, floating-point instructions are more common than integer instructions. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Yeager such that floating-point translations are more frequently used than integer translations, which would then result in the first table holding frequently used translations and the second table holding less-frequently used translations.

13. Referring to claim 7, Yeager has taught a method as described in claim 6. Yeager has further taught that said storing less-frequently used translations includes identifying said less-frequently used translations from a set of logical register addresses. Clearly, if integer register addresses and translations are encountered/identified, then the integer alias table will be used for storing.

14. Referring to claim 8, Yeager has taught a method as described in claim 7. Yeager has further taught that said identifying includes selecting infrequently used temporary registers. See column 14, lines 56-57. Basically, with renaming, the next free registers physical registers will be used in translation. These next free registers are infrequently used (less used) with respect to the previous free registers which have already been used. In addition, physical register contents are temporary (i.e., they are overwritten by new instructions).

15. Referring to claim 9, Yeager has taught a method as described in claim 8. Yeager has further taught that said infrequently used temporary registers are associated with a long micro-operation flow. All registers are associated with long programs (i.e., they are available for use by the long program).

16. Referring to claim 10, Yeager has taught a method as described in claim 7. Yeager has further taught that said identifying includes selecting control registers. All registers in the system may be considered control registers because they control the data that is used during execution. For instance, say an addition instruction is of the form ADD R1, R2, R3. The numbers added will be controlled by R2 and R3.

17. Referring to claim 11, Yeager has taught a method as described in claim 10. Yeager has further taught that said identifying includes choosing registers used by a compiler. Column 1, lines 50-52, allude to the existence of a compiler, which translates high-level code into low-level code such as assembly/machine code. The compiler uses all registers in the translation of high-level to low-level code. For instance, Z = X+Y in high level code could be translated into:

```
LD R1 X          //load value of X into register R1
LD R2 Y          //load value of Y into register R2
ADD R3 R1 R2     //add R1 and R2 and store result in R3, which represents Z
```

So, it can be seen that with enough code to translate, registers are selected by the compiler in the translation process.

18. Referring to claim 12, Yeager has taught a method as described in claim 6. Yeager has not taught building a trace in a trace cache whose micro-operations require no more live-in registers and live-out registers using said second register alias table than a first number of read ports of said second register alias table. However, Official Notice is taken that a trace cache and

its advantages are well known and accepted in the art. Trace caches speed up the fetching process by caching instruction traces, which are sequences of decoded instructions. For example, a taken branch penalty is eliminated, since separate basic blocks appear contiguous in a trace. Also, since decoded instructions are stored in the trace cache, there is no need to decode them again next time they are fetched, which would save time. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Yeager to include a trace cache that provides instructions to the first and second RATs. And, one would be motivated to make such a combination to achieve time savings in the fetching/decoding process. In a given trace, each micro-operation will require at most 1 live-out register (Fig.3, component 372) and 2 live-in registers (components 368 and 370). Therefore, each micro-operation requires at most three registers, which is less than a number of read ports of the second table. Recall that the second table has 12 read ports.

19. Referring to claim 13, Yeager has taught a method as described in claim 12. The modified Yeager has further taught that said building includes permitting no more live-out registers using said second register alias table than a second number of write ports of said second register alias table. Again, any group of 4 instructions in a given trace cannot create more than 4 live-out registers because there are only 4 write ports.

20. Referring to claim 15, Yeager has taught an apparatus comprising:

a) means for storing translations from logical register addresses to physical register addresses in a first register alias table. See Fig.3, components 204 and 352 (floating-point rename table with 16 read ports).

b) means for storing translations from logical register addresses to physical register addresses in a second register alias table, where said second register alias table has fewer read ports than said first register alias table that includes at least one read port for each source operand for an instruction is greater than said second number. See Fig.3, components 206 and 354 (integer rename table with 12 read ports). Also, see column 8, lines 55-59, and note that there is a read port for each source operand for an instruction. That is, among the 16 read ports in the floating-point rename table, four of those ports are assigned to an instruction, and among those four ports, each port is assigned to an operand (source or destination).

c) Yeager has not explicitly taught that the translations stored in the first register alias table are frequently used translations while the translations stored in the second register alias table are less-frequently used translations. However, recall that Yeager's first table stores translations corresponding to floating-point operations while Yeager's second table stores translations corresponding to integer operations. Consequently, if there are more floating-point instructions in a given program than integer instructions, then the floating-point translations would be considered to be frequently used translations with respect to integer translations, which would be less-frequently used. Clearly, any given program could include more floating-point instructions than integer instructions. For instance, in signal processing (DSP) applications, floating-point instructions are more common than integer instructions. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Yeager such that floating-point translations are more frequently used than integer translations, which would then result in the first table holding frequently used translations and the second table holding less-frequently used translations.

21. Referring to claim 16, Yeager has taught an apparatus as described in claim 15. Yeager has further taught that said means for storing less-frequently used translations includes identifying said less-frequently used translations from a set of logical register addresses. Clearly, if integer register addresses and translations are encountered/identified, then the integer alias table will be used for storing.

22. Referring to claim 17, Yeager has taught an apparatus as described in claim 15. Yeager has not taught means for building a trace in a trace cache whose micro-operations require no more live-in registers and live-out registers using said second register alias table than a first number of read ports of said second register alias table. However, Official Notice is taken that a trace cache and its advantages are well known and accepted in the art. Trace caches speed up the fetching process by caching instruction traces, which are sequences of decoded instructions. For example, a taken branch penalty is eliminated, since separate basic blocks appear contiguous in a trace. Also, since decoded instructions are stored in the trace cache, there is no need to decode them again next time they are fetched, which would save time. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Yeager to include a trace cache that provides instructions to the first and second RATs. And, one would be motivated to make such a combination to achieve time savings in the fetching/decoding process. In a given trace, each micro-operation will require at most 1 live-out register (Fig.3, component 372) and 2 live-in registers (components 368 and 370). Therefore, each micro-operation requires at most three registers, which is less than a number of read ports of the second table. Recall that the second table has 12 read ports.

23. Referring to claim 18, Yeager has taught an apparatus as described in claim 17. The modified Yeager has further taught that said building includes permitting no more live-out registers using said second register alias table than a second number of write ports of said second register alias table. Again, any group of 4 instructions in a given trace cannot create more than 4 live-out registers because there are only 4 write ports.

24. Referring to claim 20, Yeager has taught a system comprising:

a) a processor including a first register alias table including a first number of read ports to translate a first set of logical register addresses to physical register addresses. See Fig.3, components 204 and 352 (floating-point rename table with 16 read ports).

b) a second register alias table including a second number of read ports to translate a second set of logical register addresses to physical register addresses, wherein said first number that includes at least one read port for each source operand for an instruction is greater than said second number. See Fig.3, components 206 and 354 (integer rename table with 12 read ports). Also, see column 8, lines 55-59, and note that there is a read port for each source operand for an instruction. That is, among the 16 read ports in the floating-point rename table, four of those ports are assigned to an instruction, and among those four ports, each port is assigned to an operand (source or destination).

c) Yeager has not explicitly taught an audio input/output device and an interface to couple said processor to said audio input/output device. However, Official Notice is taken that interfacing audio devices to a processor is well known and accepted in the art. That is processors are known to be coupled to devices such as speakers and microphones, which allow the processor to produce sounds and process sounds, such as speech. This allows for additional interaction

between the user and machine. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Yeager to include an audio I/O device which is interfaced to the processor.

25. Referring to claim 21, Yeager has taught a system as described in claim 20. Yeager has further taught that said first number is proportional to a third number of logical register addresses in said first set. See Fig.3 and column 8, lines 55-67, and note that a third number of addresses is 16 addresses (4 addresses (registers 358, 360, 362, and 364) in each of 4 parallel instructions). This is the reason 16 read ports are required (so that 4 instructions, each having 4 registers, may be renamed in parallel). Clearly, 1 set of 4 registers 4 read ports. So, it can be seen that 4 addresses is to 4 read ports, 8 addresses is to 8 read ports, 12 addresses is to 12 read ports, and 16 addresses is to 16 read ports. Consequently, the first number is proportional to a third number.

26. Referring to claim 23, Yeager has taught a system as described in claim 20. Yeager has not taught a trace cache to supply a trace of micro-operations to said first register alias table and said second register alias table. However, Official Notice is taken that a trace cache and its advantages are well known and accepted in the art. Trace caches speed up the fetching process by caching instruction traces, which are sequences of decoded instructions. For example, a taken branch penalty is eliminated, since separate basic blocks appear contiguous in a trace. Also, since decoded instructions are stored in the trace cache, there is no need to decode them again next time they are fetched, which would save time. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Yeager to include a trace cache that provides instructions to the first and second RATs. And, one would be motivated to make such a combination to achieve time savings in the fetching/decoding process.

27. Referring to claim 24, Yeager has taught a system as described in claim 23. The modified Yeager has further taught that said trace cache includes trace cache logic to build said trace limiting a third number of live-in and live-out logical registers to said second number. Each group of 4 instructions executed in parallel from a given trace can only have up to 4 live-out registers (4 read ports) and 8 live in-registers (8 read ports).

Allowable Subject Matter

28. Claims 3, 14, 19, and 22 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Response to Arguments

29. Applicant argues on pages 16-17 of the remarks that the prior art of record does not teach the claims as amended. However, the examiner disagrees and would like to direct applicant's attention to the rejections above for reasons why the amendments do not overcome the prior art of record.

Conclusion

30. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL.** See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (571) 272-4168. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

DJH
David J. Huisman
November 7, 2006

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100